

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

**APPLICATION FOR LETTERS PATENT**

\* \* \* \* \*

**METHOD OF DISCRIMINATING BETWEEN  
DIFFERENT TYPES OF SCAN FAILURES,  
COMPUTER READABLE CODE TO CAUSE A  
DISPLAY TO GRAPHICALLY DEPICT ONE OR  
MORE SIMULATED SCAN OUTPUT DATA SETS  
VERSUS TIME AND A COMPUTER  
IMPLEMENTED CIRCUIT SIMULATION AND  
FAULT DETECTION SYSTEM**

\* \* \* \* \*

**INVENTORS**

Jerome Bombal and Laurent Souef

ATTORNEY'S DOCKET NO. VL5-062

1 **METHOD OF DISCRIMINATING BETWEEN DIFFERENT TYPES OF**  
2 **SCAN FAILURES, COMPUTER READABLE CODE TO CAUSE A**  
3 **DISPLAY TO GRAPHICALLY DEPICT ONE OR MORE SIMULATED**  
4 **SCAN OUTPUT DATA SETS VERSUS TIME AND A COMPUTER**  
5 **IMPLEMENTED CIRCUIT SIMULATION AND FAULT DETECTION**  
6 **SYSTEM**

7 **TECHNICAL FIELD**

8 The present invention relates to a method of discriminating  
9 between different types of scan failures, computer readable code to cause  
10 a display to graphically depict one or more simulated scan output data  
11 sets versus time and a computer implemented circuit simulation and fault  
12 detection system.

13 **BACKGROUND OF THE INVENTION**

14 Integrated circuits have rapidly increased in complexity, operating  
15 speed and utility. One technique for specifying an integrated circuit  
16 design is with a hardware description language (HDL) such as VHDL.  
17 A hardware description language (HDL) enables representation of an  
18 integrated circuit design at a logical level, and provides a high level  
19 design language. An integrated circuit is represented in several different  
20 levels, comprising different layers of abstraction. Silicon compilers,  
21 comprising synthesis programs, are used to yield a final implementation  
22 wherein the programs generate sufficient detail to proceed directly to  
23 silicon fabrication.

1 A compiler generates a netlist of generic primitive cells during the  
2 processing of an HDL program. A netlist is a list of all the nets, or  
3 collection of pins needing to be electrically connected, in a circuit. The  
4 netlist consists of a detailed list of interconnections and logic  
5 components, and can include primitive cells such as XOR gates, NAND  
6 gates, latches and D-flip flops and their associated interconnections.

7 The silicon compiler first generates a netlist of independent cells,  
8 and then applies a particular cell library to the resulting generic netlist  
9 via a process called mapping. As a consequence, a dependent mapped  
10 netlist is generated which uses standard circuits that are available within  
11 a cell library and which are available to the computer system. Silicon  
12 compilers and mapping programs are well understood in the art, and are  
13 described in numerous patents including U.S. Patent Nos. 5,406,497 and  
14 5,831,868, which are hereby incorporated herein by reference.

15 As circuit complexity has grown, it has been increasing difficult  
16 and expensive to test functionality of integrated circuits. Strategies that  
17 have evolved to cope with this include design for testability (DFT), a  
18 feature placed into an integrated circuit whereby predetermined test  
19 control signals place the circuit into a test mode. Application of special  
20 test input signals from an automated test pattern generator (ATPG) to  
21 inputs to the integrated circuit results in a set of output signals. The  
22 output signals are compared to expected values in order to determine if  
23 the integrated circuit provided the expected values. When a discrepancy

1 is noted between the output signals and the expected values, it is  
2 necessary to determine how the discrepancy arose in order to be able  
3 to propose a repair, re-design or other remedial measure.

4 In some types of DFT, after a test signal is used to set the  
5 integrated circuit into the test mode, sequential and combinatorial logic  
6 circuits are tested by interconnecting selected flip-flops within the  
7 integrated circuit into a shift register (also known as a "scan register")  
8 in the test mode using multiplexers. A test vector that includes known  
9 input signals is applied to portions of the integrated circuit, and the  
10 resultant output signals are first captured in parallel in, and then serially  
11 clocked (or "scan shifted") out of, the scan registers.

12 It is expensive to design and manufacture new integrated circuits.  
13 It is particularly expensive to manufacture prototype integrated circuits  
14 that do not operate as expected or desired. Accordingly, it is common  
15 to simulate operation of new designs as they are being developed in  
16 order to try to identify as many potential errors or problems as possible  
17 prior to finalizing and then manufacturing the prototype design.

18 Typical simulation software tools, such as those available from  
19 Mentor Graphics (Wilsonville, OR) or Synopsys (Mountain View, CA),  
20 provide a text file output containing information regarding simulated scan  
21 shifting. A great deal of time and effort is often involved in tracing  
22 back from error flags in these text files, using netlist parsing and  
23 calculations, to determine where the problem actually lies. This process

1 is also sufficiently complex that it is error-prone, at least in part because  
2 this process fails to provide any intuitive grasp of where the problem  
3 lies.

4 Problems that may occur during simulated output signal capture  
5 include bad sampling by the flip-flop during a capture cycle, due to a  
6 race condition or other problem, improper clocking behavior and  
7 improper reset behavior, both of which latter conditions may be caused  
8 by clock signal spikes. Problems that may occur during simulated scan  
9 shifting include clock skew issues leading to data loss in the register, an  
10 unexpected reset that destroys some scan data, a missing clock pulse due  
11 to dysfunctional clock gating or interruption of the scan chain, which  
12 may be due to bad gating or multiplexing.

13 What is needed is a tool that provides an intuitive understanding  
14 of signal flow in automated simulation of new integrated circuit designs,  
15 and that promotes ready and rapid discrimination between simulated  
16 shift-induced errors and simulated signal capture errors in integrated  
17 circuit designs incorporating design for testability.

## 18 19 SUMMARY OF THE INVENTION

20 The invention provides a method of discriminating between  
21 different types of scan failures. In one aspect of the invention, the  
22 method includes simulating a scan enable signal to a circuit represented  
23 by a netlist corresponding to a scan chain of flip-flops that are coupled

1 together to form a shift register and that are also coupled to  
2 combinatorial logic being tested. The method also includes simulating  
3 initiation of a data capture cycle in the netlist corresponding to the scan  
4 chain, the data capture cycle simulating circuit operation to provide  
5 simulated output data including a series of scan flops from the scan  
6 chain being simulated together with the combinatorial logic. The method  
7 further includes simulating scanning data out from each flop in the scan  
8 chain and into a test program. The test program: extracts simulated  
9 scan flops from the simulated circuit operation data; sorts the simulated  
10 scan flops into a logical order; identifies labels for the simulated scan  
11 flops; and graphically displays the simulated scan flops versus time  
12 together with the labels.

13 In another aspect, the present invention includes an article of  
14 manufacture comprising a computer usable medium having computer  
15 readable code embodied therein to cause a display to graphically depict  
16 one or more simulated scan output data sets versus time. The computer  
17 readable program code in the article of manufacture includes a module  
18 to extract the simulated scan flops of one or more scan chains from the  
19 simulated scan output data, a module to sort the extracted simulated  
20 scan flops into a logical order, a module to identify labels for the  
21 simulated extracted scan flops and a module to graphically display the  
22 simulated scan flops versus time together with the labels.  
23

1 In yet another aspect, the present invention includes a computer  
2 implemented circuit simulation and fault detection system. The system  
3 includes memory configured to provide a database and operative to store  
4 a netlist including nets of an integrated circuit under design, an  
5 automatic test pattern generation algorithm operative to design and  
6 simulate an integrated circuit design and processing circuitry configured  
7 to simulate operation of the integrated circuit design to provide  
8 simulated circuit operation data and to identify types of defects occurring  
9 during simulation of the integrated circuit design. The processing  
10 circuitry is operative to: extract simulated scan flops from the simulated  
11 circuit operation data; sort the simulated scan flops into a logical order;  
12 identify labels for the simulated scan flops; and graphically display the  
13 simulated scan flops versus time together with the labels.

#### 14 BRIEF DESCRIPTION OF THE DRAWINGS

15  
16 Fig. 1 is a simplified schematic diagram of a circuit including  
17 combinational logic circuit and scan flip-flops, in accordance with an  
18 embodiment of the present invention.

19 Fig. 2 is a simplified schematic diagram of one of the scan flip-  
20 flops of Fig. 1, in accordance with an embodiment of the present  
21 invention.  
22  
23

1 Fig. 3 is a simplified block diagram of a computer aided design  
2 (CAD) system coupled to an integrated circuit, in accordance with an  
3 embodiment of the present invention.

4 Fig. 4 is a simplified block diagram of the design process for an  
5 integrated circuit having design-for-testability features, in accordance with  
6 an embodiment of the present invention.

7 Fig. 5 is a simplified flow chart illustrating operation of an  
8 exemplary software module for processing and displaying simulated test  
9 data corresponding to the circuit of Fig. 1 via the system of Fig. 3, in  
10 accordance with an embodiment of the present invention.

11 Fig. 6 is a simplified graph showing correct simulated test results  
12 corresponding to the circuit of Fig. 1 obtained via the system of Fig. 3  
13 using the process of Fig. 5, in accordance with an embodiment of the  
14 present invention.

15 Fig. 7 is a simplified graph showing simulated test results indicative  
16 of a shift problem corresponding to the circuit of Fig. 1 obtained via the  
17 system of Fig. 3 using the process of Fig. 5, in accordance with an  
18 embodiment of the present invention.

19 Fig. 8 is a simplified graph showing simulated test results indicative  
20 of a capture problem from the circuit of Fig. 1 obtained via the system  
21 of Fig. 3 using the process of Fig. 5, in accordance with an embodiment  
22 of the present invention.  
23



## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

This disclosure of the invention is submitted in furtherance of the constitutional purposes of the U.S. Patent Laws "to promote the progress of science and useful arts" (Article 1, Section 8).

The present invention includes methods and apparatus for streamlining analysis of simulated test results from integrated circuits that include design for testability features. More particularly, the present invention permits graphical display of simulated test results in a manner facilitating rapid and robust determination of fault location and promoting intuitive fault localization.

In the following description, numerous specific details are set forth, such as particular architecture, hardware configurations etc., in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known methods and hardware configurations are not described in detail in order to not obscure the present invention.

The present invention addresses problems encountered in simulation of testing of modern large scale integrated circuits that include design for testability features. The problems stem from the large amount of data generated during test simulation and from lack of an intuitive approach to sorting and analyzing the simulated test data.

1 Figs. 1 and 2 describe operation and construction of integrated  
2 circuits using design for testability techniques. Figs. 3 and 4 describe  
3 hardware and software used for simulating operation of such integrated  
4 circuits. Fig. 5 describes a process for interpreting the test results, and  
5 Figs. 6 through 8 show results from the process of Fig. 5.

6 Fig. 1 is a simplified schematic diagram of a circuit 20 including  
7 combinational logic circuit 22 and scan flip-flops 24, in accordance with  
8 an embodiment of the present invention. The illustrated circuit 20  
9 includes combinational logic circuitry 22 coupled with plural flip-flops 24  
10 and control circuits 26, also known as glue logic. The flip-flops 24 are  
11 individually identified as FF0-FF8 in Fig. 1. Signals from the Q outputs  
12 of the flip-flops 24 corresponding to data captured from the  
13 combinational circuitry 22 are referred to as "flops." Circuit 20 can be  
14 implemented in many configurations, such as ASICs, controllers etc.,  
15 including different combinational logic circuitry 22 in other embodiments.  
16 The illustrated flip-flops 24 comprise scan flip-flops and collectively form  
17 a shift register, known as a scan register 28. The scan register 28,  
18 together with the combinational logic 22 coupled to the scan register 28,  
19 are collectively known as a "scan chain." Other flip-flop or device  
20 configurations can be utilized in other circuit arrangements, such as level  
21 sensitive scan designs (LSSD). Examples of several types of scan chains,  
22 including those used in LSSD, are discussed in U.S. Patent No.  
23 5,920,575, entitled "VLSI Test Circuit Apparatus And Method" and

1 issued to Gregor et al., and in U.S. Patent No. 5,909,453, entitled  
2 "Lookahead Structure For Fast Scan Testing" and issued to Kelem et al.,  
3 which patents are hereby incorporated herein by reference.

4 Fig. 2 is a simplified schematic diagram of the scan flip-flops 24  
5 of Fig. 1, in accordance with an embodiment of the present invention.  
6 The scan flip-flops 24 individually include an internal multiplexer 30  
7 coupled with an internal D-type flip-flop 32. The illustrated scan flip-  
8 flop 24 includes a D input and a scan-in input Si coupled to the  
9 multiplexer 30. An output S of the multiplexer 30 is coupled to a D  
10 input of the flip-flop 32.

11 An enable signal SCANENABLE is applied to a scan-enable input  
12 Se of the illustrated scan flip-flop 24 and to the multiplexer 30 to  
13 control the application of data from either the D input or the scan-in  
14 input Si to the D input of the flip-flop 32. A clock signal can also be  
15 applied to a clock input CK of the depicted scan flip-flop 24 and to the  
16 flip-flop 32 to control the timing of operations of the scan flip-flop 32.  
17 The Q output of the flip-flop 32 forms a Q output of the scan flip-  
18 flop 24. The illustrated scan flip-flop 24 also includes a control input  
19 which comprises a reset input in the depicted illustration. Other control  
20 inputs can be provided within individual scan flip-flops 32, such as a set  
21 input, for example.

22 Scan flip-flop configurations can be utilized to provide increased  
23 flexibility in circuit design. For example, scan flip-flops 24 can be

utilized to implement test mode operations responsive to assertion of a test mode signal TESTMODE (not illustrated). The TESTMODE signal is selectively asserted by an external circuit tester (discussed below with reference to Figs. 3 and 4) in the described embodiment. For example, the circuit 20 of Fig. 1 operates in a normal, functional mode when the TESTMODE signal is a logic "0." Alternatively, the circuit 20 operates in a test mode when the TESTMODE signal is a logic "1."

The SCANENABLE signal can additionally be utilized to control operation of the scan flip-flops 24. For example, when the SCANENABLE signal is a logic "0" during the test mode, operations are provided in a capture mode. Alternatively, when the SCANENABLE signal is a logic "1," operations are provided in a scan mode, also referred to as a shift mode.

In general, the SCANENABLE signal controls the application of data from the D input or the scan-in input Si to the Q output of individual scan flip-flops 24, corresponding to operation in the capture and scan modes, respectively. Data is received into the scan flip-flops 24 from the combinational logic circuitry 22 during capture operations. Such data can be subsequently scanned through the scan register 28 and out of the FF0 flip-flop 24 during scan modes of operation. Alternatively, scan-in data is applied to the scan flip-flops 24 and thus to the combinational logic circuitry 22 during scan modes of operation.

1 When the SCANENABLE signal is a logic "0," the D input of the  
2 scan flip-flop 24 coupled to the multiplexer 30 is coupled with the D  
3 input of the flip-flop 32. When the SCANENABLE signal is a logic  
4 "1," the scan-in input Si of the scan flip-flop 24 coupled to the  
5 Multiplexer 30 is coupled with the D input of the flip-flop 32.  
6 Accordingly, normal data from the logic circuitry 22 can be selectively  
7 applied via the D input into the scan flip-flops 24. Alternatively, scan  
8 data can be selectively inputted using the scan-in inputs Si into the scan  
9 flip-flops 24.

10 Referring again to Fig. 1, the circuit 20 operates in a functional  
11 mode and a test mode as previously described. The circuit 20 operates  
12 in the functional mode during normal operation, such as with an  
13 associated device in a given application. The circuit 20 can be coupled  
14 with a circuit tester (discussed below with reference to Figs. 3 and 4)  
15 which performs testing operations during test mode operations.

16 The combinational logic circuitry 22 is coupled with individual  
17 control circuits 26. The control circuits 26 comprise OR gates in the  
18 described embodiment, corresponding to the reset inputs of the scan flip-  
19 flops 24 being active low. Alternatively, the control circuits 26 can  
20 comprise AND gates if the reset inputs of the scan flip-flops 24 are  
21 active high. Other configurations for the control circuits 26 are possible.

22 The combinational logic circuitry 22 is configured to generate  
23 control signals to control operations within respective scan flip-flops 24.

1 Exemplary operations comprise reset operations in the illustrated  
2 embodiment. In other configurations, the combinational logic circuitry 22  
3 can control other functions of associated scan flip-flops 24.

4 The control circuits 26 individually include an input to receive  
5 control signals from the combinational logic circuitry 22. The control  
6 circuits 26 are preferably configured to selectively provide such received  
7 control signals to control inputs of respective scan flip-flops 24 during  
8 testing of the circuit 20 in the test mode of operation. As described  
9 below, the control circuits 26 are also preferably operable to selectively  
10 disable the provision of control signals received from the logic circuitry  
11 22 to the respective control inputs of scan flip-flops 24 during the  
12 testing of the circuit 20. In the described embodiment, the control  
13 circuits 26 are also configured to pass the control signals received from  
14 the logic circuitry 22 to the respective control inputs of the flip-flops 24  
15 during operation of the circuit 20 in the functional mode of operation.

16 The control circuits 26 also individually include an input adapted  
17 to receive an enable signal to control the selective provision of control  
18 signals received from the logic circuitry 22 to the control inputs of the  
19 respective scan flip-flops 24 during testing of the circuit 20. An  
20 exemplary enable signal includes the RESETnENABLE signal, where the  
21 letter "n" reflects potential presence of one or more other scan chains  
22 in addition to that associated with the circuit 20.  
23

1 While individual outputs or flops from the scan flip-flops 24 are  
2 not normally accessible when an integrated circuit including the circuit  
3 20 of Fig. 1 is actually manufactured, these outputs, labeled "Flop[0]"  
4 through "Flop[8]" in Fig. 1, can be made available during simulated  
5 operation of the circuit 20. It is desirable to reduce the number of  
6 input/output pins in the integrated circuit when it is manufactured, and  
7 this is why the scan chain is configured as a shift register. This  
8 configuration allows many different outputs from the combinational logic  
9 circuitry 22 of Fig. 1 to be sequentially read from a single output pin  
10 coupled to the Q output of the FF0 scan flip-flop 24. However, the  
11 scope of simulated output data includes very large amounts of data  
12 describing simulated operation of the circuit and includes the signals  
13 "Flop[0]" ... "Flop[8]." These output data signals are generated, typically  
14 as text files, and it is extremely difficult and time-consuming to parse  
15 the circuit description together with the simulated output data to  
16 determine the nature and location of an error in the proposed design.

17 Fig. 3 is a simplified block diagram of a computer system  
18 incorporating novel aspects of the present invention and identified by  
19 reference numeral 40. The computer system 40 is configured to  
20 implement an electronic design automation (EDA) system 42 that is  
21 capable of simulating operation of a design for the circuit 20 of  
22 Figure 1. A circuit designer inputs an integrated circuit design that  
23 includes design-for-testability features, validates the design, places

1 components onto a chip layout and routes connections between  
2 components. According to one construction, an integrated circuit 46  
3 under design and test comprises an application specific integrated circuit  
4 (ASIC) 48.

5 The electronic design automation (EDA) system 42 includes a  
6 central processing unit (CPU), or processor, 50, a memory 52 and a data  
7 storage device 54, all coupled to other elements of the system 42 via a  
8 bus 47. In one form, the memory 52 comprises a random access  
9 memory 56, a read only memory 58 and a data storage device 54. In  
10 one form, the data storage device 54 comprises a hard disk drive. The  
11 CPU 50 is used to implement an operating system and application  
12 programs, such as EDA and ATPG programs. Furthermore, the CPU  
13 50 is used to implement the novel features of the present invention.

14 A human designer, user or operator inputs design information into  
15 the system 42 via a keyboard 60 and/or a cursor manipulating tactile  
16 input device 62, such as a mouse or a touchpad. However, it is  
17 understood that other forms of input devices can also be used including  
18 voice recognition systems, joysticks, graphics tablets, data readers, card  
19 readers, magnetic and optical readers, other computer systems etc. The  
20 designer receives visual feedback on the design process via an output  
21 device 64. According to one construction, the output device 64  
22 comprises a graphics display terminal, such as a CRT display or a liquid  
23 crystal display. During synthesis and testing of a design, the memory 52



1 is used to store logic design information for an integrated circuit 46  
2 under design.

3 In operation, the designer specifies the logic design of the  
4 integrated circuit 48 via a commercially available form of design capture  
5 software 76 such as software that is commercially available from  
6 Synopsys, Inc. and Cadence Design Systems, Inc. A behavior description  
7 file 78 is output from the design capture software 76. The behavior  
8 description file 78 is written in a hardware description language (HDL),  
9 such as VHDL. The behavior description file 78 represents the logic  
10 design of a proposed design at a register transfer level.

11 The behavior description file 78 provides an input to a logic design  
12 synthesis program 80, such as a VHDL design compiler 81. The logic  
13 design synthesis program 80 is operative to create circuitry and gates  
14 necessary to realize a design that has been specified by the behavior  
15 description file 78. One commercially available VHDL design compiler  
16 is sold by Synopsys, Inc. The VHDL design compiler cooperates with  
17 the logic synthesis design compiler 78 to generate a detailed description  
18 file 82. The detailed description file 82 includes a gate-level definition  
19 of the logic design for the proposed integrated circuit design. The  
20 detailed description file 82 comprises a netlist for the design under  
21 consideration.

22 The detailed description file 82 is input into several EDA system  
23 programs such as an automatic test pattern generation (ATPG) program

84, as well as placement and routing tools, timing analyzers and simulation programs. The ATPG program 84 generates test patterns that are used in the system 42 to simulate operation of a proposed design for the integrated circuit 20 of Fig. 1, using a netlist, in the form of the detailed description file 82, that is input to the ATPG program 84. In accordance with the prior art, the system 42 provides simulated data output as a text file.

Fig. 5 is a simplified flow chart illustrating operation of an exemplary software module for processing and displaying the output file including simulated test data corresponding to the circuit 20 of Figure 1 using the EDA system 42 of Figs. 3 and 4, in accordance with an embodiment of the present invention. As used herein, the term "module" includes lines of code that may or may not be defined by a subroutine separate from the main program.

Fig. 5 illustrates a process "P1" that is initiated by the designer following or concurrently with utilization of the ATPG program 84 of Fig. 4 via the computer 40 of Fig. 3. According to step "S1," the scan enable signal "SCANENABLE" of Fig. 1 is simulated. After performing step "S1," the process "P1" proceeds to step "S2."

In step "S2," the process "P1" simulates a data capture signal, such as the RESETnENABLE signal of Fig. 1. After performing step "S2," the process "P1" proceeds to step "S3."

1 In step "S3," the process "P1" simulates scan chain data using the  
2 ATPG tool. After performing step "S3," the process "P1" proceeds to  
3 step "S4."

4 In step "S4," the process "P1" extracts simulated scan flops from  
5 the simulation data output by the ATPG tool 84 of Fig. 4. The  
6 extracted scan flops may include data corresponding, for example, to the  
7 signals "Flop[0]" through "Flop[8]" of Fig. 1. After performing step  
8 "S4," the process "P1" proceeds to step "S5."

9 In step "S5," the process "P1" sorts the simulated scan flops. In  
10 one embodiment, the step "S5" involves sorting the simulated scan flops  
11 into ordered groups, with each group of scan flops corresponding to a  
12 specific scan chain in the circuit 20 being simulated. In one  
13 embodiment, the simulated signals in each group are sorted into  
14 sequential order, e.g., "Flop[0]," "Flop[1]," ... "Flop[8]." After  
15 performing step "S5," the process "P1" proceeds to step "S6."

16 In step "S6," the process "P1" identifies labels (e.g., "Flop[0],"  
17 etc.) for each of the simulated signals. After performing step "S6," the  
18 process "P1" proceeds to step "S7."

19 In step "S7," the process "P1" graphically displays the simulated  
20 signals (and their labels) selected by the designer versus time. In one  
21 embodiment, the process "P1" further displays test mode signals such as  
22 the reset enable signal RESETnENABLE or the scan enable signal  
23 SCANENABLE from Fig. 1, to facilitate interpretation of the scan flop

1 data and to enable identification of the capture cycle (see Figs. 4  
2 through 6 and associated text). In one embodiment, the process "P1"  
3 optionally also displays expected results from one of the scan flops, such  
4 as the expected scan output data from FF0 of Figure 1. After  
5 performing step "S7," the process "P1" proceeds to step "S8."

6 In step "S8," the process "P1" determines if the designer wishes  
7 to display simulated data corresponding to another scan chain. When  
8 the designer does not wish to display additional simulated data, the  
9 process "P1" ends. When the designer wishes to display additional  
10 simulated data, the process "P1" proceeds to step "S9."

11 In step "S9," the process "P1" determines which simulated data the  
12 designer wished to see displayed, and iterates steps "S4" through "S8"  
13 until the designer determines that no further simulated data need to be  
14 displayed.

15 Fig. 6 is a simplified graph showing correct simulated test results  
16 from the circuit 20 of Fig. 1 obtained via the system 42 of Fig. 3 using  
17 the process "P1" of Fig. 5, in accordance with an embodiment of the  
18 present invention. The top trace corresponds to the reset enable signal  
19 RESETnENABLE of Fig. 1, and allows the designer to identify the time  
20 period corresponding to the capture cycle (see vertical dashed lines  
21 corresponding to the RESETnENABLE signal going to logic "0"). The  
22 simulated signals "Flop[8]" through "Flop[0]" (see Fig. 1) are displayed  
23 in order below the reset signal RESETnENABLE. The bottom trace

1 corresponds to the expected scan output data, in this case from  
2 simulation of an expected ScanOut signal (Fig. 1).

3 As exemplified by the dashed arrow extending from the "Flop[8]"  
4 signal diagonally down and to the right to terminate on the "Flop[0]"  
5 signal, data corresponding to flops 8 through 0 are propagating through  
6 the scan register 28 (Fig. 1) in an orderly fashion. Note that the  
7 dashed arrow may be laterally translated and still will show correct data  
8 propagation through the scan register 28. This indicates that there are  
9 no shift problems occurring in the scan register 28 in this simulation.

10 Comparison of the "Flop[0]" signal to the expected scan output  
11 data signal also shows that the simulated data from the shift register are  
12 identical to the expected scan output data. This indicates that the  
13 simulation of the circuit 20 of Fig. 1 indicates proper operation of this  
14 scan chain in this test, and this, in turn, fails to identify any problems  
15 with simulation of the combinational logic circuitry 22 or the chain  
16 including scan register 28.

17 Fig. 7 is a simplified graph showing simulated test results indicative  
18 of a shift problem from the circuit 20 of Fig. 1 obtained via the system  
19 42 of Fig. 3 using the process "P1" of Fig. 5, in accordance with an  
20 embodiment of the present invention. The traces are organized as  
21 described above with respect to Fig. 6. In Fig. 7, comparison of the  
22 bottom two traces shows that the "Flop[0]" signal differs from the  
23 expected scan output data. This comparison, which may be carried out

1 automatically, shows immediately that some form of error has occurred  
2 in the simulation, and an error message may be generated and may be  
3 displayed to indicate that the simulation has identified an error.

4 Comparison of the "Flop[2]" and "Flop[3]" signals shows that these  
5 two signals are identical. This comparison, which may be carried out by  
6 automatically comparing each possible pair of adjacent flops to determine  
7 if any two adjacent flops are identical, is a clear indication of a shift  
8 problem. An error message may be generated and may be displayed to  
9 indicate that the simulation has identified a shift problem and to indicate  
10 which pair of flops are associated with the shift problem. In the  
11 example shown, examination of the flop[2] and flop[3] clock signals is  
12 likely to show a large skew between them. Fixing or obviating the clock  
13 skew by, for example, adding a lockup latch or balancing the clock tree,  
14 will remedy the issue. In any event, the problem has been rapidly  
15 identified without having to resort to a large text file, parsing the netlist  
16 or the like.

17 Fig. 8 is a simplified graph showing simulated test results indicative  
18 of a capture problem from the circuit 20 of Fig. 1 obtained via the  
19 system 42 of Fig. 3 using the process "P1" of Fig. 5, in accordance with  
20 an embodiment of the present invention. The traces are organized as  
21 described above with respect to Figs. 6 and 7, however, the expected  
22 scan output data trace is different, corresponding to either a different  
23 input vector or a different combinational logic circuitry 22 (Fig. 1).

1 Because the erroneous simulated signal has propagated through the scan  
2 chain (scan register 28 in Fig. 1) as shown by the arrow, the scan  
3 behavior is correct. This indicates a data capture problem occurring in  
4 the FF7 flip-flop 24 of Fig. 1, and an error message indicating that a  
5 data capture problem exists with the flip-flop 24 providing the "Flop[7]"  
6 signal. Examining the simulated input signals to the FF7 flip-flop 24 is  
7 very likely to reveal the source of the problem. This allows the problem  
8 to be traced back through the scan chain without the designer having to  
9 interpret a complex text file, without having to parse the netlist  
10 corresponding to the circuit 20 and without any calculations. As a  
11 result, the speed and accuracy with which the problem can be identified  
12 are both improved:

13 In compliance with the statute, the invention has been described  
14 in language more or less specific as to structural and methodical  
15 features. It is to be understood, however, that the invention is not  
16 limited to the specific features shown and described, since the means  
17 herein disclosed comprise preferred forms of putting the invention into  
18 effect. The invention is, therefore, claimed in any of its forms or  
19 modifications within the proper scope of the appended claims  
20 appropriately interpreted in accordance with the doctrine of equivalents.  
21  
22  
23